98. (Unchanged) A computer program product for applying a set of rules as in claim 96, wherein the step of controlling the method comprises:

instructions for executing method logic of the method.

REMARKS

Claims 1-28, 31-39, 41-52, 55-63, 65-76, 79-87 and 89-98 are pending in the present application. By this Response, claims 29, 30, 40, 53, 54, 64, 77, 78 and 88 are canceled and claims 1-2, 12, 23, 27, 36, 46, 48, 51, 60, 70, 72, 75, 84, 94 and 96 are amended. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

I. Information Disclosure Statement

The Office Action questions whether the present invention is related to FLOWMARK, a product of the present Assignee, Common Object Request Broker (CORBA), ISMOD, DevelopMate, Newi, and Object Management Workbench. Applicants have submitted documents which were known to Applicants and which Applicants believe to be material to patentability. Applicants are under no obligation to provide information regarding references first raised by the Examiner. If the Examiner believes that these products are relevant to the present application, the Examiner must cite the necessary references in a PTO Form 892. It is not Applicants' duty to research references first raised by the Examiner. Thus, Applicants are under no duty to search for references of which they are unaware - it is the Examiner's duty to perform a search of the prior art and cite those references that the Examiner believes to be material to the patentability of Applicants' claimed invention. Applicants' only duty is to disclose references of which they are aware that are material to patentability.

II. Objection to the Specification

The Office Action objects to the specification as including an incorrect serial number for one of the cross referenced U.S. Patent applications. By this Response, Applicants have amended the specification to correct the error in the U.S. Patent application serial number. Accordingly, Applicants respectfully request withdrawal of the objection to the specification.

III. Examiner's Interpretation

In this section of the Office Action, the Examiner sets forth the Examiner's "interpretations" of the terms "modeling" and "Flow Control." The Examiner's "interpretations" of these terms in no way limits Applicants' claimed invention. The terms in the claims must be interpreted in light of the specification as one of ordinary skill in the art would interpret these terms.

To the contrary, the Examiner in supposedly "interpreting" the term "modeling" only refers to sections of the Martin reference, discussed hereafter. The Examiner does not interpret the term "modeling" in light of the present specification. Thus, the claims are not bound by the Examiner's alleged "interpretation."

Furthermore, with regard to the "interpretation" of the term "Flow Control", the Examiner admits that Applicants do not claim flow control. Thus, the claims cannot be limited by the Examiner's "interpretation" of the term "Flow Control." With regard to whether what is recited in the claims may be "a product" of "flow control," this is immaterial to whether the claims as a whole are directed to patentable subject matter. Flow control is not claimed and thus, the claims should not be limited to the Examiner's interpretation of the meaning of flow control.

With regard to the Examiner's "interpretation" of the term "decorating," the present specification on page 4 clearly describes a mechanism by which a decorator pattern is used to add a new behavior to an object to thereby generate a "decorated" object. Thus, the term "decorating" in claim 3 should be interpreted in light of the specification, not the personal interpretations of the Examiner.

In summary, none of the Examiner's personal interpretation of the terms above may be used to limit the scope of the pending claims for the reasons noted above.

IV. 35 U.S.C. § 102, Anticipation

The Office Action rejects claims 1-98 under 35 U.S.C. § 102 as being anticipated by Martin, "Principles of Object-Oriented Analysis and Design," published June 1, 1992. This rejection is moot with regard to the canceled claims and is respectfully traversed with regard to the remaining claims.

With regard to claim 1, the Office Action states:

> Martin anticipates a computer implemented process for applying a set of rules (Martin, Chapter 10, RULES, and page 138-139 and 249-251), the process comprising:
>
> (a) placing a pre-method control before logic of a method (Martin, page 142, operation precondition) and post method control point after the logic of the method (Martin, page 142, post condition)
>
> (b) associating a set rules with each control point (Martin, page 142, 147 "Operation" as per (a) above) based on a class of object in which the method resides (Martin, page 143, "...rules associated with diagrams of OO..."), name of the method and type of control point, whether the pre-method control point or the post-method control point (Martin, page 142, operation precondition);
>
> (c) invoking the method (Martin, page 116), wherein encountering each control point during the execution of the method comprises (Martin, page 142, post condition)
>
> (i) determining if the encountered control point is active (Martin, page 122, IF structure in center diagram);
>
> (ii) on the basis of an active control point (Interpreted as the result of the IF structure above further described in Appendix A on page 381 Control Conditions);

1) selecting rules based on a set of rules associated with the active control point associated in step (Martin, page 122, first diagram example is the control condition to fire missiles)(b);

2) running the selected rules (Martin, page 122, rule that lead to the control condition);

3) obtaining results from running the rules (Martin, page 122, trigger rule at the bottom of the page); and

4) combining the results using a combining algorithm specified by the control point (Martin, page 122, A control condition can function as a combining algorithm as seen in diagram in middle of the page and page 126 Figure 9.9 and Martin teaches a way to have a combining algorithm where one of three operations are selected as on page 124, and Martin teaches a way to have a combining algorithm where on can be selected as taught in the mutually exclusive notation on the bottom of the page 125).

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that Martin does not identically show each and every feature of the pending claims arranged as they are in the claims. Furthermore, Applicants respectfully submit that the claims do not read on the processes specifically described in the Martin reference and that the Office Action is engaged in applying broad teachings of Martin without regard for the actual teachings of Martin or the specific features recited in the claims.

Chapters 9 and 10 of Martin, which are the basis for the Office Action's rejection of all of claims 1-98, teach a method of modeling the behavior of an object oriented system. The modeling involves representing operations, preconditions of the operations, post conditions of the operations, control conditions, events and triggers. In the modeling described by Martin, an operation may have a precondition that identifies what must happen before the operation executes, and a post condition that describes the result of the

operation if the operation executes with the precondition being satisfied. The operation may further include a control condition which is a condition that is used to determine whether the operation is to execute or not. The control condition must be checked prior to invoking the operation and may be a complex collection of Boolean conditions (page 122). Martin further teaches the implementation of these preconditions, post conditions and control conditions as rules in Chapter 10. Martin does not teach "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point," "determining if the encountered control point is active," and "selecting rules based on a set of rules associated with the active control point associated in step (b)," as recited in claim 1.

With regard to the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point," the Office Action alleges that this feature is taught by Martin on pages 142, 143, and 147. However, these sections of Martin have nothing to do with associating a set of rules with a control point based on a class of object in which the method resides, name of the method and type of control point. Page 142 merely describes that the event diagram is an executable diagram from which program code may be generated using a tool such as the OO-CASE tool. Page 143 merely describes that rules may be either object state rules or object behavior rules; that rules may be used with other types of diagrams other than event diagrams; and that rules may be stated as English expressions which may then be used to generate code. Page 147 merely shows examples of an operation, event diagram and state transition diagram that have rules attached to them.

There is nothing in these sections, or any other sections, of Martin that can remotely be considered to teach the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point." It is not clear where or how the Office Action can extract any teaching from Martin that even suggests the features of the presently claimed invention. Rather, it appears that the Office Action is engaged in taking broad teachings of event diagrams and generating code from event diagrams, and reading into these broad teachings the very specific features recited in Applicants' claims. However, the Office

Action cannot read into the prior art features that are only present in Applicants' own disclosure and use the "modified" prior art to support a rejection of the claims. The Office Action must base its rejection solely on the teachings of the references themselves. In this case, there is nothing in the Martin reference that can be used to support a position of anticipation with regard to the above feature.

Similarly, there is nothing in the Martin reference that remotely even hints at the feature of "determining if the encountered control point is active." The Martin reference does not even recognize a possibility of having active or inactive control points. The Office Action equates the operation precondition and post conditions to the control points recited in the claims, even though they are not the same as the recited control points. A control point, as defined in the present specification is a point at which rules may be attached to add additional functionality. The preconditions and post conditions of Martin are merely requirements for the operation to execute properly.

However, assuming that the preconditions and post conditions are the same as a control point, arguendo, Martin describes the precondition and post conditions as always having to be satisfied in order for proper operation of the object oriented system model. Thus, the precondition and post conditions are always "active" and there is no need to determine if they are active enough. It is for this reason that Martin does not mention anywhere in the entire reference, any step of determining if a control point is active.

The Office Action alleges that this feature is taught in Martin at page 122 simply because Martin teaches an IF structure. The Office Action equates the precondition with the control point and then states that the presence of an IF structure in the precondition is the same as determining if a control point is active. This does not make any sense since, as is clearly described in the present invention and recited in the claims, if a control point is not active, the rules of the control point are not executed. If the precondition of Martin were the same as a control point, then the precondition must always be active in order for the IF structure to even operate. Therefore, the IF structure would always indicate that the precondition is active and thus, there is no need for the IF structure. Thus, the IF structure is not the same as the step of determining if a control point is active. Again, the Office Action is reading in teachings from the present specification into general

descriptions provided by Martin with out any regard for what is actually being claimed and what Martin actually teaches.

Just as with the above, Martin also provides no teaching that is remotely similar to the feature of "selecting rules based on a set of rules associated with the active control point associated in step (b)." While Martin teaches that rules may be used to implement the precondition and post condition of an operation, there is no teaching in Martin of associating a set of rules with an active control point and then selecting rules based on this set of rules. Martin provides no teaching at all regarding selection of rules and the Office Action has not pointed out with particularity any section of Martin that teaches a selection of rules based on a set of rules associated with the active control point.

The Office Action alleges that this feature is taught by Martin in the first diagram on page 122 (although the following text indicates that the Examiner is in actuality referring to the last figure on page 121). These diagrams merely illustrate the use of control conditions that are checked prior to an operation executing. There is no selection of rules even mentioned or shown in these figures, let alone the selection of rules based on a set of rules associated with an active control point. The Office Action is referencing portions of the Martin reference that do not even have anything to do with the features of the claim.

It is not understood how the figures referred to can be interpreted in any way to teach the selection of rules based on a set of rules associated with an active control point. The figure illustrating firing of a missile is provided to show how a control point may be used to handle a plurality of triggers that are required for the operation to execute. The first diagram on page 122 also illustrates this concept. However, nowhere in the figures of the accompanying text is there anything mentioned about selection of rules based on a set of rules associated with an active control point.

On page 122 Martin does state that the control condition may be a collection of "or" conditions applied to triggers (see the middle diagram on page 122). However, the "or" Boolean condition does not constitute a selection of rules based on a set of rules associated with an active control point. Rather, the "or" conditions merely operate to state that if any one of the triggers satisfies the control condition, then the operation will execute. There is no selection of rules.

Thus, Martin does not teach each and every feature of claim 1 as is required under 35 U.S.C. § 102(b). Accordingly, Applicants respectfully request withdrawal of the rejection of claim 1 under 35 U.S.C. § 102(b).

With regard to claim 2, Martin does not teach "defining a control point just before logic of at least one method, wherein the control point is a method context control point." A method context control point is a control point that may have a variety of different rules and different types of rules associated with it as the need for these rules changes over time (see page 11, lines 17-29 of the present specification). Martin does not teach such method context control points. Rather, Martin teaches preconditions, post conditions, and control conditions that perform specific functions that are not intended to change over time and are not intended to have different rules and different types of rules associated with them. This is because these conditions are linked to the operation so closely that the conditions cannot be changed without causing the operation to not execute properly.

The method context control point of the present invention allows the object model to be modified without having to change the underlying model itself. By applying new rules at active control points, new functionality is available without having to modify the model. With the event diagrams of Martin, the model itself must be modified, i.e. the operations and their preconditions, post conditions and control conditions, in order to obtain new functionality. Thus, the present claims do not read on the Martin reference and the Martin reference does not anticipate the presently claimed invention.

Since Martin does not teach method context control points, Martin does not teach each and every feature recited in claim 2 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 2, Martin also does not teach or suggest the features recited in dependent claims 3-11. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 2-11 under 35 U.S.C. § 102(b).

In addition to the above, Martin does not teach many of the features recited in the dependent claims 3-11. For example, Martin does not teach decorating an object to dynamically insert a first control point, as recited in claim 3. As mentioned above, "decorating" as described by the present specification is a method of using a decorator pattern to generate a decorated object. Martin does not teach or even mention inserting a

control point dynamically, let alone by decorating the object. The Office Action alleges that this feature is taught by Martin at pages 59-60 where it is stated "Every time we tell an advanced CASE tool about classes, inheritance, and so on, it should generate code." However, this statement has nothing to do with dynamically inserting a control point into an object, let alone "decorating" the object. Rather, this statement merely says that when a class is generated, code should be generated for the class.

Martin also does not teach the step of "externalizing the at least one control point as a class and instantiating it at the at least one control point," as recited in claim 6. The Office Action equates the precondition and post condition of an operation as the same as a control point. However, the precondition and post condition of an operation are not a class and are not instantiated. The precondition and post condition of Martin are tied to an operation, which may be represented as a method. Thus, the precondition and post condition cannot be classes and cannot be instantiated.

The Office Action alleges that this feature is taught at pages 133-136 and page 167. However, there is nothing in these, or any other sections, of the Martin reference that remotely comes close to a control point that is externalized as a class which is instantiated at a control point. The Office Action is again engaged in reading into Martin teachings that simply are not there and are only present in Applicants' disclosure.

Martin also does not teach "defining, with a control point, at least one of a rule selecting algorithm and a rule results combination algorithm," as recited in claim 10. The Office Action does not even state where this teaching may be found in Martin but merely refers to the rejection of claim 1. There is nothing in Martin that provides for a selection algorithm that is defined for a control point. This is because rules cannot be selected for a precondition, post condition or control condition. Rather, the rules are assigned to these conditions and are fixed thereafter. There is no selection algorithm for selecting amongst rules that are associated with a condition in Martin.

With regard to claim 11, Martin does not provide for changing of rules associated with a control point in a set of rules. The Office Action alleges that this feature is taught by Martin simply because it teaches the use of OO-CASE and I-CASE. However, when rules are assigned to a precondition, post condition or control condition, as taught by Martin, they are fixed and are not changeable without destroying the proper functioning

of the associated operation. This is because the precondition must be satisfied in order for the operation to execute and the post condition must be satisfied in order for the operation to execute properly. Thus, if these are changed, the operation will execute improperly.

Claims 12, 46, 48, 51, 70, 72, 75, 94, 96 define over Martin for similar reasons as noted above with regard to claim 2. In particular, Martin does not teach a method context control point as recited in each of these claims and as discussed above. At least by virtue of their dependency on claims 12, 46, 48, 51, 70, 72, 75, 94 and 96, Martin does not teach the features recited in dependent claims 13-15, 49-50, 52-59, 73-74, 76, 79-83, 97-98. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 12-15, 46, 48-59, 72-74, 75-76, 79-83, 94 and 96-98 under 35 U.S.C. § 102(b).

In addition to the above, Martin also does not teach many of the features recited in the dependent claims set forth above. For example, Martin does not teach that the rules perform at least one function that varies over time, as recited in claim 14. As mentioned above, the precondition, post condition and control condition are only used to determine whether an operation is to execute and whether an operation has executed correctly. There is nothing in Martin that teaches a control point that has rules that perform a function that varies over time. None of the functions of the precondition, post condition or control condition in Martin vary over time.

The Office Action alleges that this feature is taught by Martin at page 117 simply because Martin teaches clock events and that Martin teaches associating rules with event diagrams. The clock events have nothing to do with preconditions, post conditions and control conditions. The clock events are separate entities in the event diagram from preconditions, post conditions and control conditions. The Office Action is again taking general teachings and conjuring up specific teachings regardless of the actual teachings of Martin. The only place that control points having rules that perform functions that vary over time are even mentioned is in Applicants' own disclosure. While clock events are triggered based on time, this in no way teaches a control point that has a rule that performs a function which varies over time.

With regard to claims 49, 73 and 97, Martin does not teach controlling a method by exiting the method. As noted above, the operation in Martin may be implemented as a

method. For the method to be invoked, the precondition must be satisfied and the control condition must be satisfied. Thus, the only options are that the operation is either executed because both the precondition and the control conditions are satisfied or the operation is not executed because one or both of the precondition and control condition are not satisfied. There is no possibility of "exiting" the operation based on the precondition and the control condition because the operation is not entered until after the precondition and control condition operate. Thus, Martin does not teach controlling the method by exiting the method as described in claims 49, 73 and 97.

Regarding claims 55 and 79, Martin does not teach flagging means for flagging the at least one control point on the basis of being active. As previously mentioned, Martin does not even recognize that a control point may be active or non active. Thus, Martin cannot flag a control point based on whether it is active.

The Office Action merely points to its rejection of claim 1 in rejecting this feature. However, none of the sections cited in the rejection of claim 1, nor the rejection itself, address anything remotely resembling flagging a control point on the basis of the control point being active. Martin does not even discuss active and non active control points.

With regard to claims 56, 58, 80 and 82, Martin does not teach defining a rule selection algorithm associated with the at least one control point. As previously mentioned above, Martin does not teach any mechanism for selecting rules, let alone a rule selecting algorithm that is associated with a control point. The Office Action alleges that Martin teaches this feature at page 168. However, there is nothing on page 168, or in any other section of Martin, that teaches or even remotely suggests a rule selection algorithm that is associated with a control point. The Martin reference merely teaches the association of rules with event diagram preconditions, post conditions and control conditions. There is no selection mechanism associated with these preconditions, post conditions and control conditions.

With regard to independent claim 16, Martin does not teach defining rules to the at least one control point on the basis of the object's class name, method name, and position of the at least one control point in the method. The lack of teachings in Martin

with regard to this feature are discussed in detail in the rejection of claim 1 above and the claim 16 defines over Martin for these same reasons.

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claim 16 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 16, Martin also does not teach the features recited in dependent claims 17-22. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 16-22 under 35 U.S.C. § 102(b).

In addition to the above, Martin also does not teach many of the specific features set forth in dependent claims 17-22. For example, Martin does not teach activating at least one control point having associated rules, as recited in claim 17. As discussed in depth above, Martin does not even recognize the possibility of having active and nonactive control points, let alone activating a control point.

Similarly, this lack of teaching in Martin applies to claim 22 which recites "deactivating the at least one control point." Again Martin does not teach active or nonactive control points and thus, cannot teach deactivating a control point.

With regard to claims 23 and 27, Martin does not teach a method context control point, as previously discussed in detail above. The rejection of claims 23 and 27 is overcome based on the same reasoning as set forth above with regard to claims 2, 12, 46, 48, 51, 70, 72, 75, 94 and 96.

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claims 23 and 27 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claims 23 and 27, respectively, Martin also does not teach the features recited in dependent claims 24-26, 28 and 31-35. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 23-28 and 31-35 under 35 U.S.C. § 102(b).

In addition to the above, Martin does not teach many of the features recited in the dependent claims 24-26, 28 and 31-35. For example, Martin does not teach flagging the at least one control point on the basis of being active, as recited in claim 31. This feature was discussed in detail above with regard to claims 55 and 79.

Martin also does not teach a rule selection algorithm associated with at least one control point as recited in claims 32 and 34. This feature was discussed in detail above with regard to claims 56, 58, 80 and 82.

With regard to claims 36, 60 and 84, Martin does not teach associating a rule with another method within an object class. As previously mentioned above, Martin only teaches associating rules with preconditions, post conditions and control conditions in an event diagram. There is no teaching in Martin that the same rule may be associated with two different methods in an object class. While similar rules may be associated with different conditions in an event diagram, there is nothing in Martin that teaches that the same rule may be associated with two methods within an object class.

The Office Action alleges that this feature is taught on pages 266-268. The Office Action alleges that this section of Martin teaches associating rules with more than one object, reuse and inheritance which can all be read on the claimed feature. However, the general teachings of reuse and inheritance do not in themselves teach the specific features recited in claims 36, 60 and 84. There is nothing in Martin that teaches or even suggests to one of ordinary skill in the art that the same rule can be associated with a plurality of methods, as recited in claims 36, 60 and 84. The Office Action is again engaged in reading into general teachings, the specific teachings found only in Applicants' disclosure without any basis for such reading in of teachings in the reference itself.

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claims 36, 60 and 84 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claims 36, 60 and 84, Martin also does not teach the features recited in dependent claims 37-39, 41, 61-63, 65, 85-87 and 89. Accordingly, Applicants respectfully request withdrawal of the rejection of claim 36-39, 41, 60-63, 65, 84-87 and 89 under 35 U.S.C. § 102(b).

Regarding claim 42, Martin does not teach the feature of determining if a control point is active. This feature has been address above with regard to claim 1 and thus, claim 42 defines over Martin for similar reasons as noted above with regard to this feature in claim 1.

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claim 42 as is required under 35 U.S.C. § 102(b). At least by virtue of

their dependency on claim 42, Martin does not teach the features recited in claims 43-44. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 42-44 under 35 U.S.C. § 102(b).

In addition to the above, Martin also does not teach the specific features recited in claims 43-44. For example, Martin does not teach a selecting algorithm associated with an active control point, as discussed at length above.

This same distinction over Martin also applies to independent claim 45 and thus, claim 45 is allowable over the Martin reference for this same reason. With regard to claim 46, Martin does not teach a method context control point, as discussed above. Regarding claim 47, Martin does not teach determining if a control point is active. Thus, Applicants respectfully request withdrawal of the rejections of claims 45-47 under 35 U.S.C. § 102(b).

With regard to claim 66, Martin does not teach determining if a control point is active, as previously discussed above with regard to claim 42 and others. Thus, Martin does not teach each and every feature recited in claim 66 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claim 66, Martin also does not teach the features recited in dependent claims 67 and 68. In addition, Martin does not teach a selecting algorithm as recited in claim 67. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 66-68 under 35 U.S.C. § 102(b).

With regard to claim 69, again Martin does not teach a selecting algorithm associated with a control point. Regarding claim 70, Martin does not teach a method context control point. With regard to claim 71, Martin does not teach determining if control points are active. Thus, Martin does not teach each and every feature of claims 69-71. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 69-71 under 35 U.S.C. § 102(b).

Regarding claims 90, 92 and 95, Martin does not teach determining if a control point is active. With regard to claims 91 and 93, Martin does not teach a selecting algorithm. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 90-93 and 95 under 35 U.S.C. § 102(b).

V. Conclusion

It is respectfully urged that the subject application is patentable over Martin and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: January 30, 2002

Stephen J. Walder, Jr.
Reg. No. 41,534
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

APPENDIX OF AMENDMENTS

January 29, 2002

IN THE SPECIFICATION:

The redacted paragraph on page 1, lines 7-20 reads as follows:

The present invention is related to applications entitled "Method And Apparatus For General Integrity Rule Checking Point In An Application," U. S. Patent Application No. _____ , Attorney Docket No. AT9-98-267, filed even date hereof, assigned to the same assignee; "System and Method And Data Processing System For Specifying And Applying Rules To Classification-Based Decision Points In An Application System," U. S. Patent Application No. _____, Attorney Docket No. AT9-98-287, filed even date hereof, assigned to the same assignee; and "Method and Apparatus for Identifying Applicable Business Rules," U. S. Patent Application No. 09/993,718, Filed 12/18/97, assigned to the same assignee and all of which are incorporated herein by reference.

IN THE CLAIMS:

Please cancel claims 29, 30, 40, 53, 54, 64, 77, 78 and 88 without prejudice or disclaimer.

Please amend claims 1-2, 12, 23, 27, 36, 46, 48, 51, 60, 70, 72, 75, 84, 94 and 96 as follows:

1. (Once Amended) A computer implemented process for applying a set of rules, the process comprising:

(a) placing a pre-method control point before logic of a method and post-method control point after the logic of the method;

(b) associating a set of rules with each control point based on a class of object in which the method resides, name of the method, and type of control point, whether the pre-method control point or the post-method control point;

(c) invoking the method, wherein encountering each control point during the execution of the method comprises:

(i) determining if the encountered control point is active;

(ii) on the basis of an active control point:

1) selecting rules based on a set of rules associated with the active control point associated in step (b);

2) running the selected rules;

3) obtaining results from running the rules; and

4) combining the results using a combining algorithm specified by the control point.

2. (Once Amended) A computer implemented process for applying a set of rules comprising:

(a) defining an object;

(b) defining at least one method in the object;

(c) defining a control point just before logic of at least one method, wherein the control point is a method context control point; and

(d) associating a set of rules with the control point.

12. (Twice Amended) A computer implemented process for applying a set of rules, comprising:

(a) invoking a method in an object;

(b) encountering an active control point during the invocation of the method, wherein the active control point is an active method context control point;

(c) selecting rules associated with the method of the object at the control point;

(d) invoking the rules; and

(e) combining results from invoking the rules.

23. (Once Amended) A computer implemented process for applying a set of rules, comprising:

(a) defining an object;

(b) defining a method in the object;

(c) defining a first control point of the method, the first control point being a method context control point;

(d) determining rules associated with the first control point;

(e) defining a second control point of the method, the second control point also being a method context control point; and

(f) determining rules associated with the second control point.

27. (Once Amended) A computer implemented process for defining an object comprising: defining an object;

defining a method in the object by:

defining method logic;

placing the method logic in the method;

defining at least one control point, wherein the at least one control point is a method context control point; and

placing the at least one control point in the method wherein the method logic is continuous, wherein the step of placing the at least one control point further comprises placing the at least one control in the method after the method logic.

36. (Once Amended) A computer implemented process for defining a rule comprising:

creating the rule;

associating the rule with an object class;

associating the rule with a method within the object class;

associating the rule with an occurrence of a control point within the method; and

associating the rule with another method within the object class.

46. (Once Amended) A computer implemented process for applying a set of rules, comprising:

selecting an object class;

selecting a method within the object class;

invoking the method;

processing rules comprising:

encountering a method context control point;

finding at least one rule associated with the method context control point;

running the at least one rule;

determining results on the basis of running the at least one rule;

accessing a combining algorithm associated with the method context control point; and

combining the results using the combining algorithm.

48. (Once Amended) A computer implemented process for applying a set of rules, comprising:

selecting an object class;

selecting a method within the object class;

invoking the method;

processing rules comprising:

encountering a control point associated with the method, the control point being a method context control point for which associated rules may be changed from a first set of rules to a second set of rules different from the first set of rules;

finding at least one rule associated with the control point prior to executing method logic of the method;

running the at least one rule;

obtaining results on the basis of running the at least one rule; and

controlling the method on the basis of the results.

51. (Once Amended) A data processing system for defining an object comprising:

defining means for defining an object;

defining means for defining a method in the object by:

defining means for defining method logic;

placing means for placing the method logic in the method;

defining means for defining at least one control point, wherein the at least one control point is a method context control point; and

placing means for placing the at least one control point in the method wherein the method logic is continuous, wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method after the method logic.

60. (Once Amended) A data processing system for defining a rule comprising:

creating means for creating the rule;

associating means for associating the rule with an object class;

associating means for associating the rule with a method within the object class; [and]

associating means for associating the rule with an occurrence of a control point within the method; and

associating means for associating the rule with another method within the object class.

70. (Twice Amended) A data processing system for applying a set of rules, comprising:

selecting means for selecting an object class;

selecting means for selecting a method within the object class;

invoking means for invoking the method;

processing means for processing rules comprising:

encountering means for encountering a control point, the control point being a method context control point for which associated rules may be changed from a first set of rules to a second set of rules different from the first set of rules;

finding means for finding at least one rule associated with the control point;

running means for running the at least one rule;

determining means for determining results on the basis of running the at least one rule;

accessing means for accessing a combining algorithm associated with the control point; and

combining means for combining the results using the combining algorithm.

72. (Twice Amended) A data processing system for applying a set of rules, comprising:

selecting means for selecting an object class;

selecting means for selecting a method within the object class;

invoking means for invoking the method;

processing means for processing rules comprising:

encountering means for encountering a control point associated with the method, the control point being a method context control point;

finding means for finding at least one rule associated with the control point prior to executing method logic of the method;

running means for running the at least one rule;

obtaining means for obtaining results on the basis of running the at least one rule; and

controlling means for controlling the method on the basis of the results.

75. (Once Amended) A computer program product embodied on a computer readable medium containing instructions for a computer implemented process for defining an object, the instruction comprising:

instructions for defining an object;

instructions for defining a method in the object by:

instructions for defining method logic;

instructions for placing the method logic in the method;

instructions for defining at least one control point, wherein the at least one control point is a method context control point; and

instructions for placing the at least one control point in the method wherein the method logic is continuous, wherein the step of placing the at least one control point further comprises placing the at least one control in the method after the method logic.

84. (Once Amended) A computer program product embodied on a computer readable medium containing instructions for a computer implemented process for defining a rule, the instruction comprising:

instructions for creating the rule;

instructions for associating the rule with an object class;

instructions for associating the rule with a method within the object class;

instructions for associating the rule with an occurrence of a control point within the method; and

instructions for associating the rule with another method within the object class.

94. (Once Amended) A computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising:

instructions for selecting an object class;

instructions for selecting a method within the object class;

instructions for invoking the method;

instructions for processing rules comprising:

instructions for encountering a control point, the control point being a method context control point for which associated rules may be changed from a first set of rules to a second set of rules different from the first set of rules;

instructions for finding at least one rule associated with the control point;

instructions for running the at least one rule;

instructions for determining results on the basis of running the at least one rule;

instructions for accessing a combining algorithm associated with the control point; and

instructions for combining the results using the combining algorithm.

96. (Once Amended) A computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising:

instructions for selecting an object class;

instructions for selecting a method within the object class;

instructions for invoking the method;

processing rules comprising:

instructions for encountering a control point associated with the method, the control point being a method context control point;

instructions for finding at least one rule associated with the control point prior to executing method logic of the method;

instructions for running the at least one rule;

instructions for obtaining results on the basis of running the at least one rule; and

instructions for controlling the method on the basis of the results.